

Lightweight semantic prototyper for conceptual modeling

Gayane Sedrakyan and Monique Snoeck

Katholieke Universiteit Leuven,
Department of Decision Sciences and Information Management,
Naamsestraat 69, 3000 Leuven

E-mail: {gayane.sedrakyan, monique.snoeck}@kuleuven.be

(PREPRINT)

Abstract. While much research work was devoted to conceptual model quality validation techniques, most of the existing tools in this domain focus on syntactic quality. Tool support for checking semantic quality (correspondence between the conceptual model and requirements of a domain to be engineered) is largely lacking. This work introduces a lightweight model-driven semantic prototyper to test/validate conceptual models. The goal of the tool is twofold: (1) to assist business analysts in validating semantic quality of conceptual business specifications using a fast prototyper to communicate with domain experts; (2) to support the learning perspective of conceptual modeling for less experienced modelers (such as students or novice analysts in their early career) to facilitate their progression to advanced level of expertise. The learning perspective is supported by providing automated feedback that visually links the test results to their causes in the model's design. The effectiveness of the tool has been confirmed by means of empirical experimental studies.

Keywords. Conceptual modeling, semantic quality, prototyping, testing, validation, feedback

1 Introduction

Recent research highlights the need of shifting testing of a system to the earliest phases of engineering in order to reduce the time and resources spent on building a software application by minimizing the number of errors resulting from miscommunicated and/or wrongly specified requirements. Conceptual modeling is described as the process of formally describing a problem domain for the purpose of understanding and communicating system requirements [1]. Formalization of requirements through models is known to enable quality control at a level that is impossible to reach with requirements articulated in natural language. Being a sub-discipline of requirements engineering (as a means of communicating requirements) and software engineering (by providing a foundation for building information systems) [2] makes conceptual models the earliest formally testable artefact. However, despite the amount of research devoted to conceptual model quality validation methods, most of the existing

tools in this domain focus on syntactic quality. Tool support for checking semantic quality (the level to which the statements in a model reflect the real world in a valid and complete way) is largely lacking.

It is commonly accepted that prototyping is capable of achieving the most concrete form of a prospective system allowing early feedback from stakeholders thus contributing to improved semantic quality. Current methodologies for the fast prototyping of highly abstract specifications include Model Driven Engineering (MDE) of the OMG, which uses UML -the widely accepted modeling language standard [3]. However methodologies to make the MDE approach easily applicable to conceptual modeling are lacking or not (well-)defined, e.g. a way of combining structural and behavioral views into a single model. The absence of a methodology for an integrated approach combining conceptual modeling and MDE-based simulation, and prototyping in particular, also implies semantic compliance issues (is it the right prototype for a given design?). Moreover, many researchers point out the technical complexity of UML to identify a relevant subset for conceptual modelling as well as being not precise enough for rapid prototyping. Finally, a prototype's testing results are not easily interpretable since the link between a prototype and its design is not explicit. In this demo we introduce an integrated environment that allows fast prototyping of conceptual models the testing effectiveness of which is enhanced by automated feedback.

2 MERODE prototype: benefits of integrated approach

The prototyping tool introduced in this work is an integrated part of the MERODE [4] conceptual modeling environment¹. MERODE is based on the framework that overcomes the shortcomings of UML (being technically too complex for conceptual modelling and not precise enough for rapid prototyping). The framework is based on three kinds of model views (restricted class diagrams, interaction model, state charts) and uses a "consistency by construction" approach that completes missing model elements automatically thus allowing simple transformations [5, 6, 7]. A generated prototype supports the execution of business events by also providing textual and graphical feedback when and why the execution of a triggered business event is refused, thus making the links between a prototype and its design explicit. We opt for the MERODE approach for the following benefits:

- using a limited subset of UML relevant for conceptual modeling that allows removing or hiding details irrelevant for a conceptual modeling view
- framework combining structural and behavioral views into a single model
- providing integrated environment for modeling and simulation
- models designed are readily transformable for execution

The prototyping feature introduced in this work differs from similar approaches by introducing:

- easy and fast transformation using one-click approach

¹ <http://merode.econ.kuleuven.ac.be/mermaid.aspx>

- intuitive graphical interface (both for a prototyping tool and generated prototypes)
- an embedded data storage that can be queried via simple user interface invoked from inside a prototype application
- use of cognitive feedback in a prototype allowing for less experienced modelers to benefit from easy discovery of semantic compliance issues resulting from misinterpreted use of modeling language constructs

The lightweight prototyper will be of benefit to practitioners, researchers, educators and novice modelers (such as students and junior business analysts) in the domain of conceptual modeling.

3 Demonstration Highlights

We demonstrate how semantic validation of models can be facilitated by the use of a feedback-enabled prototyper. A short demo using a simple model can be viewed at <http://merode.econ.kuleuven.ac.be/demo.aspx>. A visitor to our demonstration will be invited to assume a role of a tester. S/he will be able to test the prototype of a pre made model and will be able to create and test an own model. The demo will focus on

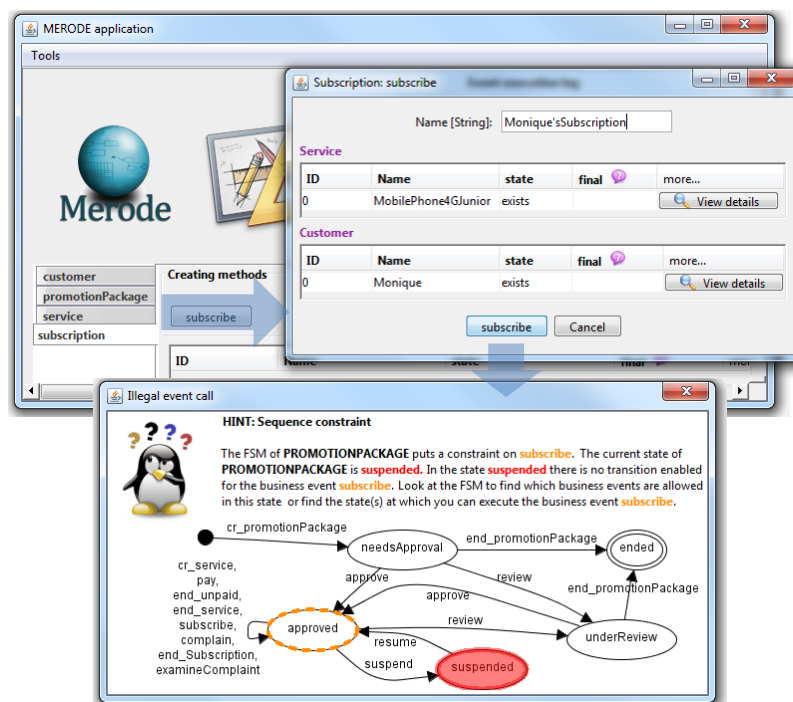


Fig. 1. An example of testing with a prototype: execution refusal due to a sequence constraint

the model-based feedback features and their effectiveness in model-understanding and validation. An example of testing an erroneous model is shown in **Fig. 1** by means of

a model about (mobile phone) services which customers can subscribe to, and for which promotion packages are offered regularly. Testing the prototype reveals a semantic mismatch (design error): trying to subscribe to a service results in execution failure due to a sequence constraint violation (the state of the PromotionPackage object to which the chosen service is associated is “suspended”). The scenario fails because of a behavioral constraint, but it actually reveals a wrong hidden dependency from “Service” to “PromotionPackage”: it seems a service depends on the availability of a promotion, which is incorrect. For a less experienced modeler the explanation can be extended with graphical visualization linking to a specific part of a model design that causes the error. Extensive experimental testing has demonstrated the positive effect of prototype-based simulation on junior modelers understanding of a model [8].

4 Conclusion

The MERODE semantic prototyper provides a fast and easy way of validating semantic quality of conceptual models thus enabling a shift of testing of a prospective system into earlier phases of its engineering, namely, conceptual specification level. Furthermore, for educational perspective the tool provides an innovative method of teaching conceptual modeling allowing novice modelers to test a working prototype of a conceptual design and learn by experiencing [9]. The integrated feedback facilitates fast progression to more advanced levels of expertise. Effectiveness of the method over traditional manual inspection and validation was confirmed by empirical experimental studies using an experimental sample of 120 users.

5 References

1. Siau, K. (2004). Informational and Computational Equivalence in Comparing Information Modeling Methods. *Journal of Database Management (JDM)*, 15(1), 73-86.
2. Moody D. L., Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions, *Data & Knowledge Engineering*, Volume 55, Issue 3, December 2005, pp. 243-276, ISSN 0169-023X.
3. OMG. (2014). Model-Driven Architecture. <http://www.omg.org/mda/>
4. Snoeck, M., Dedene, G., Verhelst, M., Depuydt, A.: Object-oriented enterprise modelling with MERODE, Leuvense Universitaire Pers, Leuven (1999)
5. Sedrakyan, G., & Snoeck, M. (2013). A PIM-to-Code requirements engineering framework. In *Proceedings of Modelsward 2013-1st International Conference on Model-driven Engineering and Software Development-Proceedings* (pp. 163-169).
6. Sedrakyan, G., & Snoeck, M. (2013). Feedback-enabled MDA-prototyping effects on modeling knowledge. In *Enterprise, Business-Process and Information Systems Modeling* (pp. 411-425): Springer.
7. Sedrakyan, G., & Snoeck, M. (2012). Technology-enhanced support for learning conceptual modeling. In *Enterprise, Business-Process and Information Systems Modeling* (pp. 435-449). Springer Berlin Heidelberg.
8. Sedrakyan, G., Snoeck, M., Poelmans, S., Assessing the effectiveness of feedback enabled simulation in teaching conceptual modeling, in review at *Computers & Education*.
9. European Commission. (2013). Opening up education: Innovative teaching and learning for all through new technologies and open educational resources.